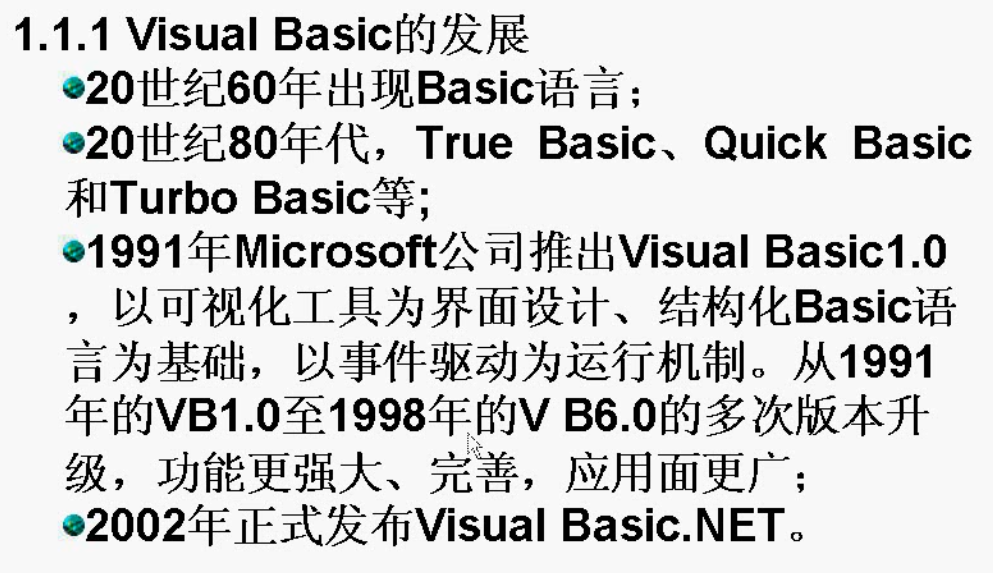
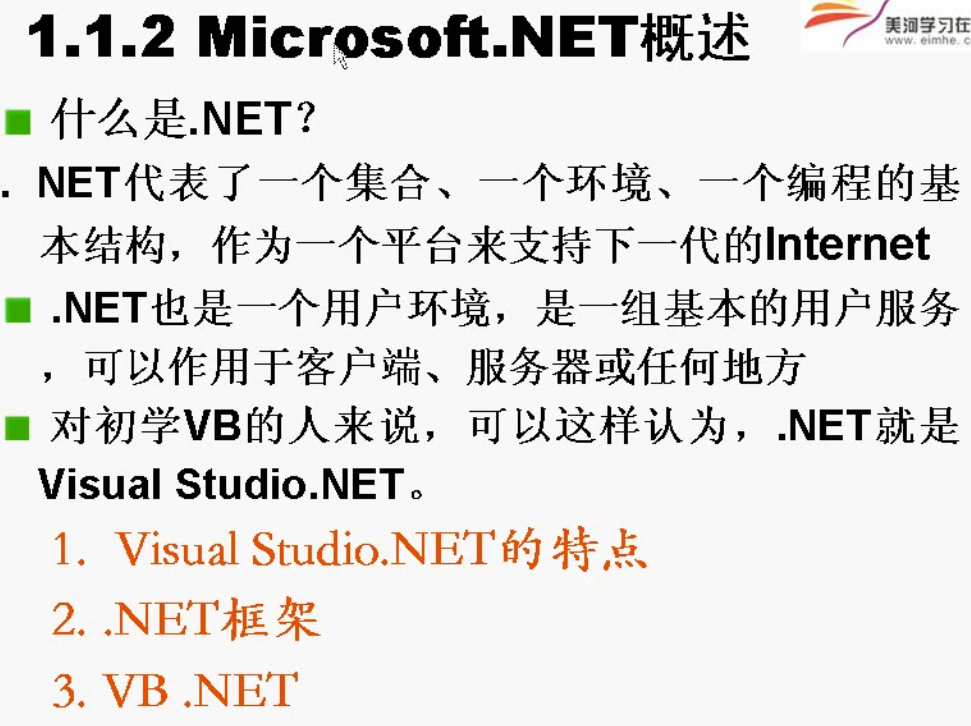
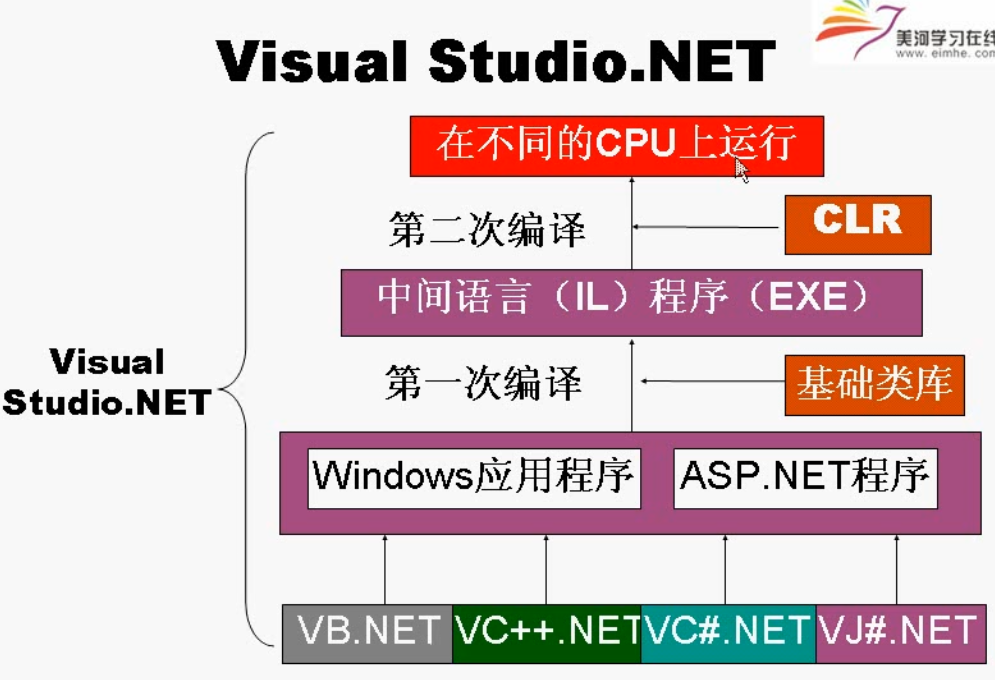
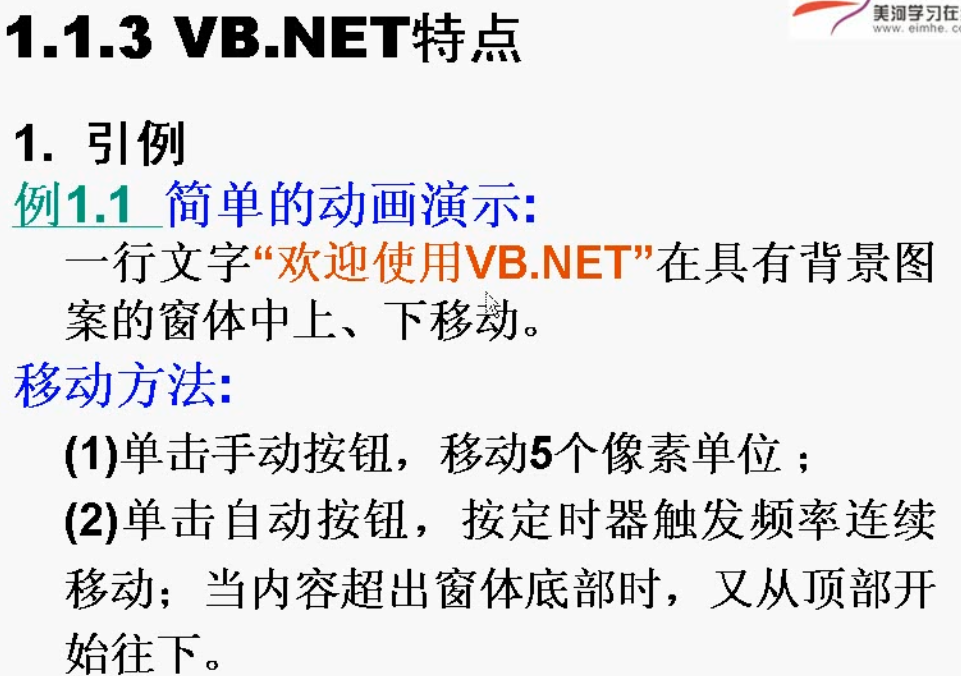
# VB.NET学习

# Visual Basic 发展









# 实际案例

## ListBox的数据绑定

我自己做的工具，可以将Array类型数据和Map类型的数据直接绑定到ListBox中

|  |
| --- |
| Public Class ListBoxUtils  ''' <summary>  ''' 通过Array的方法来设定我们的ListBox中的数据  ''' </summary>  ''' <param name="box">ListBox 插件</param>  ''' <param name="key">数组对象，ListBox中隐藏的数据，key</param>  ''' <param name="val">数组对象 listBox 中显示的数据</param>  ''' <remarks></remarks>  Public Sub setDateByArray(ByRef box As ListBox, ByVal key As Array, ByVal val As Array)  '将strtemp作为value,将strname作为text  box.Items.Clear() '清空列表框中的内容  Dim tb As New DataTable '构建一个数据表  tb.Columns.Add("1") '数据表增加两列  tb.Columns.Add("2")  '循环添加数据  For i = 0 To 2  Dim r As DataRow = tb.NewRow() '增加行  r("1") = key(i)  r("2") = val(i)  tb.Rows.Add(r)  Next i  box.DataSource = tb  box.ValueMember = "1" '绑定value  box.DisplayMember = "2" '绑定text  End Sub  ''' <summary>  ''' 通过Map的的方法来设定我们的ListBox中的数据  ''' </summary>  ''' <param name="box">ListBox 插件</param>  ''' <param name="map">我们定义的Map集合</param>  ''' <remarks></remarks>  Public Sub setDateByMap(ByRef box As ListBox, ByVal map As Hashtable)  '将strtemp作为value,将strname作为text  box.Items.Clear() '清空列表框中的内容  Dim tb As New DataTable '构建一个数据表  tb.Columns.Add("1") '数据表增加两列  tb.Columns.Add("2")  '定义Map的Entry  Dim entry As DictionaryEntry  For Each entry In map  Dim r As DataRow = tb.NewRow() '增加行  r("1") = entry.Key  r("2") = entry.Value  tb.Rows.Add(r)  Next  box.DataSource = tb  '绑定数据  box.ValueMember = "1" '绑定value  box.DisplayMember = "2" '绑定text  End Sub  End Class |

调用的方法

|  |
| --- |
| '定义两个数据  Dim key() As String = {"1", "2", "3"}  Dim val() As String = {"name", "age", "val"}  '通过Array来设定数据  Dim utils As ListBoxUtils = New ListBoxUtils()  utils.setDateByArray(lstAll, key, val)  'utils.setDateByMap(lstAll, map) |

## DatagrideView使用

栏目

## 窗口的加载和退出

|  |
| --- |
| '显示窗口2  Form2.Show()  '然后关闭我们的窗口  'Me.Close()  'Me.Dispose(True) |

### 窗口的最大化和最小化

|  |
| --- |
| '去掉上面的边框，游戏的时候，我们不需要上面的边框  'Me.FormBorderStyle = Windows.Forms.FormBorderStyle.None  '设定最大化  Me.WindowState = FormWindowState.Maximized |

## 类的初始化

### 没有参数的构造函数

|  |
| --- |
| Public Class Form2  ‘我们初始化函数  Sub New()  '初始化组建  InitializeComponent()  '然后处理数据  MsgBox("xx")  End Sub  End Class |

### 带参数的初始化函数

带参数的初始化函数

|  |
| --- |
| Public Class Form2  ‘我们初始化函数  Sub New()  '初始化组建  InitializeComponent()  '然后处理数据  MsgBox("xx")  End Sub  End Class  ’调用带有参数的函数  '初始化我们的函数  Dim f As Form2 = New Form2("逗比")  f.Show()  '然后关闭我们的窗口  'Me.Close()  'Me.Dispose(True)  End Sub |

## 类的初始化

|  |
| --- |
| Public Class User  '用户明  Public Property username As String  '用户的别名  Public Property nickname As String  '用户的密码  Public Property password As String  '用户的密码  Public Property phone As String  ''' <summary>  '''  ''' 这个是不带够着参数的 初始化数据  ''' </summary>  ''' <remarks></remarks>  Sub New()  MsgBox("数据")  End Sub  ''' <summary>  ''' 定义我们的初始化函数  ''' </summary>  ''' <param name="username"></param>  ''' <param name="nickname"></param>  ''' <param name="password"></param>  ''' <param name="phone"></param>  ''' <remarks></remarks>  Sub New(ByVal username As String,  ByVal nickname As String,  ByVal password As String,  ByVal phone As String)  Me.nickname = nickname  Me.username = username  Me.password = password  Me.phone = phone  End Sub  ''' <summary>  ''' 获取数据  ''' </summary>  ''' <remarks></remarks>  Public Sub getInfo()  MsgBox(Me.username + "\_\_" +  Me.nickname + "\_\_" +  Me.password + "\_\_" +  Me.phone + "\_\_")  End Sub  End Class |

调用的方法

|  |
| --- |
| '带有够着函数的数据  Dim user As User = New User("黄聪", "yellowcong", "yellowcong", "18604600631")  user.getInfo()  '通过没有参数，然后通过后面的数据来获取  Dim user2 As User = New User()  With user2  .username = "xx"  .phone = "xx"  .nickname = "xxx"  .password = "xx"  End With  user2.getInfo() |

## 接口的使用

|  |
| --- |
| Public Interface UserService  ''' <summary>  ''' 通过id来更新数据  ''' </summary>  ''' <param name="id"></param>  ''' <returns></returns>  ''' <remarks></remarks>  Function update(ByVal id As Integer)  End Interface |

接口实现

|  |
| --- |
| Public Class UserServiceImpl  Implements UserService  Public Function update(id As Integer) As Object Implements UserService.update  MsgBox("x子类数据" + CStr(id))  End Function  End Class |

接口的实现

|  |
| --- |
| '调用接口  Dim service As UserService = New UserServiceImpl()  service.update(22) |